A Statistical Geometric Framework for Reconstruction of Scene Models

Anastasios Manessis, Adrian Hilton, Phil McLauchlan and Phil Palmer
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford GU2 7XH, UK

Abstract

This paper addresses the problem of reconstructing surface models of indoor scenes from sparse 3D scene structure captured from N camera views. Sparse 3D measurements of real scenes are readily estimated from image sequences using structure-from-motion techniques. Currently there is no general method for reconstruction of 3D models of arbitrary scenes from sparse data. We previously introduced an algorithm for recursive integration of sparse 3D structure to obtain a consistent model. In this paper we focus on incorporating uncertainty information into model to achieve reliable reconstruction of real-scenes in the presence of noise. A statistical geometric framework is described that provides a unified approach to probabilistic scene reconstruction from sparse or even dense 3D scene structure.

1 Introduction

An important problem in computer vision is the reconstruction of 3D models of complex rigid scenes from monocular image sequences. Previous research aimed at constructing 3D models has addressed the problem of reconstruction from dense 3D surface measurements captured using active range sensors [1, 12, 2] or multi-baseline stereo [4]. Volumetric techniques have been widely used to achieve reliable reconstruction of complex objects [1] and environments [2, 11]. Methods for reconstruction from dense data assume that the distance between adjacent surface measurements can be used to estimate the local topology of the 3D surface. This assumption is not valid for interpolation of sparse 3D data.

Model reconstruction from sparse 3D data of arbitrary geometry scenes is an open problem. Faugeras et al.[3] addressed this problem using 3D Delaunay triangulation (tetrahedralisation) of a set of image features together with their visibility for each camera view to construct a volumetric model. The principal limitation of this approach is the assumption that the entire feature is visible which prohibits partial occlusion. Furthermore, this is a batch method which requires all the 3D structure prior to reconstruction.

Recently Kutulakos and Seitz [5] presented a general theory of N-view shape recovery. The principal assumption of their approach is that a locally computable consistency criteria is available to test point correspondence in multiple views. In image sequences of real-scenes such as indoor environments lack of surface texture will result in a reconstruction which deviates considerably from the real surface .

In this paper we address the problem of reconstructing surface models from sparse 3D scene structure captured from N camera views. In particular we focus on incorporat-

ing uncertainty into the reconstruction. A geometric theory that provably converges to a correct reconstruction of the real surfaces in the 3D scene as the number of processed views increases has recently been presented [6]. In this paper we extend this approach by introducing an algorithm that explicitly considers the errors inherent to a real vision system. The algorithm presented provides a unified approach to scene reconstruction from any available sparse or even dense 3D scene structure.

2 Real Scene Reconstruction

The goal of our work is to develop an automatic system for scene reconstruction from image sequences. In this section we present the algorithm developed for scene reconstruction from a sequence of images.

Images are captured using a camera mounted on an autonomous mobile robot platform. This system captures a sequence of images of an indoor scene with approximately known camera positions. A recursive structure-from-motion (SFM) algorithm [9] is applied to estimate the 3D location and uncertainty for the sparse scene features together with the camera position and orientation. A sparse feature based SFM algorithm has been used for computational efficiency of reconstruction for long image sequences. Point and line features are used in this work although the approach also extends to higher order features. The SFM algorithm incorporates constraints between features such as coplanarity and surface perpendicularity to increase reconstruction accuracy if information on feature groupings is available. Further details of this system are provide in [10].

2.1 Algorithm Overview

From each frame that SFM processes, a set of sparse 3D features $F = \{f_i\}_{i=0}^{N_f}$ is computed. Each of these features f_i which is visible in the j^{th} view taken at position $\vec{v}_j$ defines a visibility constraint c_{ij} as follows:

Definition 1 (Visibility Constraint :) The space between the view position $\vec{v}_j$ and the scene feature f_i is not occupied by an (opaque) object.

The real 3D scene viewed in frame j can be approximated by a set of planar triangular surface primitives $M = \{t_i\}_{i=1}^{N_{ts}}$ which span the space between the features. We can then define a consistent model as a set of triangles such that none of its triangles intersect any of the visibility constraints in $C = \{c_i\}_{i=0}^{N_c}$. For a single view a consistent model can be constructed by a constrained triangulation in a plane orthogonal to the view direction as the order of feature projections in the plane is preserved with respect to their relative ordering in 3D space.

In general however, for multiple views of a 3D scene there is no single 2D plane to which the scene features can be injectively projected without reordering of the features F. The algorithm developed is thus based on recursive integration of the set of feature data F_i, and feature visibility constraints C_i, for each new camera view. The algorithm can be summarised by the following steps:

1. Build an initial model M_0 by using constraint triangulation on the set of features F_0 reconstructed from the first view.

2. For each new view $i, i > 0$

(a) Update the 3D position of features in M_{i-1} for which a new measurement has been computed resulting in M_i'.

(b) Build a consistent model for the i^{th} view M_i'', by constrained triangulation of the visible features F_i, in a plane orthogonal to the view direction.

(c) Integrate non-redundant triangles from M_i'' into M_i' yielding to M_i.

(d) Eliminate triangles in M_i that violate the viewpoints visibility constraints C_i

For a closed scene with a finite set of features we have shown [6] that this algorithm will converge to a reconstruction of the real scene surfaces as the number of views increases. An underlying assumption in this algorithm however is that no significant noise should exist in the 3D data. However, in our system noise may be caused by several sources, including camera calibration, robot odometry, feature extraction, matching and 3D reconstruction. It is thus essential to use estimates of the uncertainty for our 3D measurements in order to produce a reliable system.

3 Reconstruction with Uncertainty

This section presents a probabilistic framework that utilises uncertainty on our geometric features to make our system more robust. In particular we focus on the update (2a) and visibility (2d) processes of our algorithm, as described in the previous section, that appear as the most sensitive to noisy measurements.

First we describe the underlying geometric probability assumptions and we give an uncertainty representation for both 3D points and lines. Based on this foundation we subsequently extend the update and the visibility processes to explicitly consider noise in the measurement estimates.

3.1 Geometric Uncertainty

An estimated geometric object can be considered as a random variable described by a vector **p** which consists of the variables that we have chosen to parameterise it. Thus we can define its probability density function (pdf) f(**p**) as the probability of the specific object **p** in the corresponding parameter space. In this sense geometric uncertainty can be treated using classic probability theory.

In practice we do not have an explicit pdf because we are not able to model all the sources of errors. However, a reasonable assumption is that the pdf is Gaussian. This assumption can be justified if noise is caused by a large number of independent sources from the central limit theory. There is also a practical justification for choosing the Gaussian distribution such that it can be fully specified by the first and second order statistics which are the only information that we want to propagate through the system.

This characteristic is very useful because the transformation of a pdf reduces to that of transforming its mean and covariance as any linear transformation of a Gaussian random vector is Gaussian as well [14]. In particular, if **p** is a Gaussian random vector with mean $\overline{\mathbf{p}}$ and covariance matrix P_p and assume a transformation **x=Ap+b** of **p** then **x** will be a Gaussian vector with

$$\begin{aligned} \overline{\mathbf{x}} &= \mathbf{A}\overline{\mathbf{p}} + \mathbf{b} \\ \mathbf{P_x} &= \mathbf{A}\mathbf{P_p}\mathbf{A^T} \end{aligned} \tag{1}$$

A property of geometric uncertainty is that a physical representation can be given to random variables. For a 3D point its covariance can be visually described with an ellipsoid. Assume a point $\mathbf{x}$ with covariance matrix $\mathbf{P_x}$ and mean $\overline{\mathbf{x}}$. Then it can be shown [15] that $(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}\mathbf{P_x^{-1}}(\mathbf{x}-\overline{\mathbf{x}}) = \mathbf{k^2}$ is an ellipsoid centred at $\overline{\mathbf{x}}$ that bounds the volume inside which we expect x to lie with a probability specified by k.

In our system a 3D line is defined by its endpoints $\mathbf{x_1}$, $\mathbf{x_2}$ combined to a vector $\mathbf{x} = (\mathbf{x_1 x_2})^{\mathbf{T}}$. However we are using a minimal representation with 4 degrees of freedom [8]. This results in a 3x3 covariance matrix (for each point) with the null space along the line direction. The uncertainty over each endpoint can be represented with a 2D ellipse in the plane perpendicular to the line orientation.

Having this representation we want to get an estimate of uncertainty for each point $\mathbf{p}$ along the line segment $\mathbf{x}$. To achieve this we can use a linear interpolation scheme. We can then represent $\mathbf{p}$ as

$$\mathbf{p} = \mathbf{x_1} + \lambda(\mathbf{x_2} - \mathbf{x_1}) \tag{2}$$

If we think of equation (2) as a linear transformation of $\mathbf{x}$ in the form $\mathbf{p} = \mathbf{A}(\mathbf{x})$ where

$$A = \begin{bmatrix} 1-\lambda & 0 & 0 & \lambda & 0 & 0 \\ 0 & 1-\lambda & 0 & 0 & \lambda & 0 \\ 0 & 0 & 1-\lambda & 0 & 0 & \lambda \end{bmatrix} = ((1-\lambda)I \mid \lambda I) \tag{3}$$

then according to (1) the mean and covariance $(\overline{\mathbf{x}}, \mathbf{P_p})$ will be

$$\begin{aligned} P_p &= \begin{bmatrix}(1-\lambda)I & \lambda I\end{bmatrix} \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix}(1-\lambda)I \\ \lambda I\end{bmatrix} = (1-\lambda)^2 P_1 + \lambda^2 P_2 \\ \overline{\mathbf{x}} &= (\mathbf{1}-\lambda)\overline{\mathbf{x}}_\mathbf{1} + \lambda\overline{\mathbf{x}}_\mathbf{2} \end{aligned} \tag{4}$$

where P_1 and P_2 are the covariance matrices of $\mathbf{x_1}, \mathbf{x_2}$ respectively.

Equation (4) is quadratic relative to λ which means that the uncertainty envelope around a 3D line can be visualised as an elliptic hyperboloid.

3.2 Model update based on uncertainty

The sources of error throughout the image capture and the SFM result in noisy input 3D data to our modelling system. Our reconstruction process is recursive and for every new image we update each feature in the existing model based on the new observations. This section presents how uncertainty can be utilised to make feature update robust to noise.

The initial step of the update process involves identifying outliers. Our approach is to check that each model's 3D line lies inside the uncertainty envelope of the new line estimate. In this way we check that both the distance and the relative orientation of the two lines are consistent. Another criterion used [15] requires that both 3D segments share a common part. These tests are sufficient to ensure that no false measurements are propagating to the feature integration process. Consider the case of Figure 1 where $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2})$ is the new estimate for the line and $\mathbf{p} = (\mathbf{p_1}, \mathbf{p_2})$ is the model line. The initial steps of the algorithm can then be summarised as

1. Project $\mathbf{p_i}$ onto the nearest point on the model line defined by $\mathbf{x}$ to obtain $\mathbf{x_p} = (\mathbf{x_{p1}}, \mathbf{x_{p2}})$.
2. Check that segments $\mathbf{x}, \mathbf{x_p}$ share a common part.

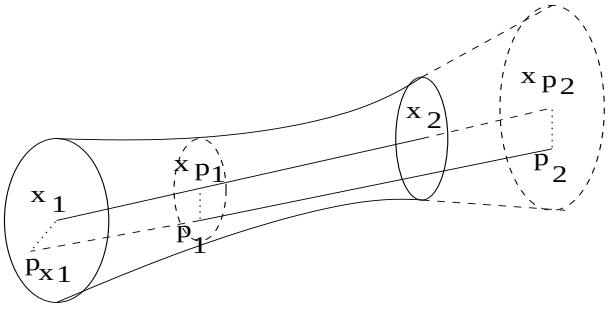

Figure 1: Feature update process.New line measurement $\mathbf{x}$ checked for consistency and combined with corresponding model line $\mathbf{p}$

3. Compute the uncertainty ellipse for each $\mathbf{x_{pi}}$ using equation (4).

4. Test whether or not $\mathbf{p_i}$ lies inside the ellipse of the corresponding $\mathbf{x_{pi}}$ point. ($\mathbf{p_i}$ lies on the same plane with the ellipse because $\overrightarrow{p_i x_{pi}} \cdot \overrightarrow{x_1, x_2} = \mathbf{0}$)

Each new consistent line measurement should subsequently be combined with any existing corresponding line estimate and integrated to the global 3D model. There are two requirements that this process should meet. First the covariance at the endpoints of the resulted line should be less than the covariance of the endpoints of the lines from which it has originated. Second the new line should not be smaller than either of the two integrated segments.

The approach used is based on the Kalman filter equations and is similar to [13] applied for merging parallel affine transformations under uncertainty. Considering again the case of Figure 1 assume that $\mathbf{l}$ is the result of merging $\mathbf{p}, \mathbf{x}$. Also assume that $\mathbf{P_l}, \mathbf{P_p}$ and $\mathbf{P_x}$ are their corresponding covariances and $\overline{\mathbf{l}}, \overline{\mathbf{p}}$ and $\overline{\mathbf{x}}$ their means. Then the Kalman gain will be $\mathbf{K} = \mathbf{P_x} * (\mathbf{P_x} + \mathbf{P_p})^{-1}$ and the covariance and mean of $\mathbf{l}$ are:

$$\begin{aligned} \mathbf{P_l} &= \mathbf{P_x} - \mathbf{K} * \mathbf{P_x} \\ \overline{\mathbf{l}} &= \overline{\mathbf{x}} + \mathbf{K} * (\overline{\mathbf{p}} - \overline{\mathbf{x}}) \end{aligned} \tag{5}$$

If **x,p** are Gaussian distributed with white noise then equation (5) is the maximum likelihood estimate with variance less than any other linear unbiased estimate [7]. The integration part of the feature update algorithm is

1. Identify the greatest segment that can be formed from $\mathbf{x_i}, \mathbf{x_{pi}}$. ($\mathbf{k} = (\mathbf{x_1}, \mathbf{x_{p2}})$).

2. Shift each of the two intermediate points $(\mathbf{x_{p1}}, \mathbf{x_2})$ along their corresponding line until they coincide with the endpoints of $\mathbf{k}$ $(\mathbf{p_1} \rightarrow \mathbf{p_{x1}}, \mathbf{x_2} \rightarrow \mathbf{x_{p2}})$.

3. Extrapolate the covariances for $\mathbf{p_{x1}}, \mathbf{x_{p2}}$ using equation (4).

4. Combine segments $(\mathbf{p_{x1}}, \mathbf{p_2})$ and $(\mathbf{x_1}, \mathbf{x_{p2}})$ using equation (5).

3.3 Visibility test with uncertainty

The visibility of features from our current camera position relative to the reconstructed model is a powerful tool for testing the consistency of our model. However, noise in the data can result in rejection of hypothesised triangles which actually correspond to real scene surfaces. This makes the original algorithm 'brittle' in the presence of noisy measurements. Thus visibility should be applied with caution and consideration to the uncertainty of both the model's triangles and features.

Our reconstruction scheme described in section (2.1) is based on the assumption that for each frame i the resultant model M_i is consistent. Each feature f_i in frame i defines a visibility constraint which should be tested against M_i. To account for the uncertainty in the measurement data we give a new definition for the visibility check that eliminates triangles which we are 'confident' that violate the test not due to noise.

Definition 2 (Visibility constraint with uncertainty) : The space between the camera position and the volumetric uncertainty envelope of the feature should not intersect with any of the model's triangles uncertainty volumes.

In this section we present the process of applying visibility for a 3D point against the reconstructed model. Visibility for lines is a direct extension for each of their two endpoints. The first step is to test for 2D overlaps between the feature and the model's triangles in the image plane. Any identified triangles should then be checked for visibility. Consider the case of Figure 2(a) where point $\mathbf{p_i}$ lies inside triangle $\mathbf{t_i}$ in the image plane while $\mathbf{p}, \mathbf{t}$ are their 3D estimates respectively. For the case of line visibility we perform the same 2D test for both of its endpoints and to each identified triangle we associate two entries which corresponds to either the intersection point of the 2D line with one of the triangle's edges or the line's endpoint which is bounded by the triangle.

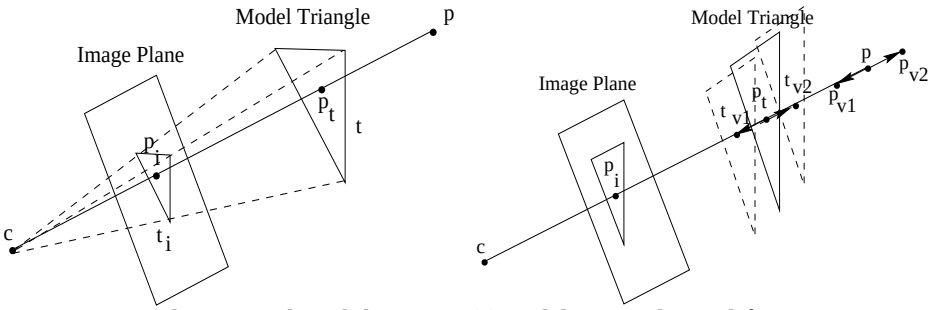

Figure 2: One dimensional visibility test. (a)Model's triangles and features are tested for 2D overlap in the image plane. (b)Representation of 1D variance of $\mathbf{p}, \mathbf{p_t}$ along the projection optical ray.

Implementation of the visibility checks according to Definition 2 involves intersections between higher order surfaces in 3D for which it is also very difficult to compute an explicit equation. Therefore, we approximate this test by reducing the search space to 1D by 'projecting' the uncertainty on the 3D line from the camera to the feature. Consider the 3D line $\mathbf{r}$ originating at the camera centre $\mathbf{c}$ and passing through $\mathbf{p_i}$. If we backproject $\mathbf{r}$ against the plane formed by the 3D triangle $\mathbf{t}$ we get point $\mathbf{p_t}$. Collinear to this 3D line is also the 3D point $\mathbf{p}$ (Figure 2(b)). If $\mathbf{p_v} = (\mathbf{p_{v1}}, \mathbf{p_{v2}})$ is the segment on $\mathbf{r}$ which depicts the variance of $\mathbf{p}$ along this line and respectively $\mathbf{t_v} = (\mathbf{t_{v1}}, \mathbf{t_{v2}})$ the segment for $\mathbf{p_t}$ then

Definition 3 (One dimensional visibility test with uncertainty) : The line segment between the camera position $\mathbf{c}$ and the furthest estimate for the position where the feature lies $\mathbf{t_{v2}}$ should not overlap with any segment projection of a model triangle's variance onto the optical ray line.

The problem thus reduces to the computation of the 1D variances for both $\mathbf{p}$ and $\mathbf{p_t}$. However, while the distribution $(\overline{\mathbf{p}}, \mathbf{P_p})$ for $\mathbf{p}$ is known $\mathbf{p_t}$ can be any point inside triangle $\mathbf{t}$. In the following section we describe a representation for triangle uncertainty and how to interpolate the distribution of any point inside it from the distributions of its vertices. A method for computing the mean and variance along a specific 1D (line) direction in space is subsequently presented.

3.3.1 Triangle Uncertainty Representation

A triangle can be fully described by its vertices. Using barycentric coordinates we can interpolate any point that lies inside the triangle. Assume triangle $\mathbf{t} = (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3})$ and a point $\mathbf{p_t}$ on the triangle. We can then express $\mathbf{p_t}$ in barycentric coordinates as

$$\mathbf{p_t} = \mathbf{u}\mathbf{x_1} + \mathbf{v}\mathbf{x_2} + (\mathbf{1} - \mathbf{u} - \mathbf{v})\mathbf{x_3} \tag{6}$$

where $\mathbf{u}, \mathbf{v} \in [\mathbf{0}, \mathbf{1}]$. If we treat each of the triangle's vertices as 3D points with known mean and covariance then following the same reasoning we used in section (3.1) for lines we can express the mean and covariance $(\overline{\mathbf{p}}_\mathbf{t}, \mathbf{P}_{\mathbf{p_t}})$ of every point on this triangle relative to the mean and covariances of its vertices $(\overline{\mathbf{x}}_\mathbf{i}, \mathbf{P}_{\mathbf{x_i}})$ as:

$$\begin{aligned} \overline{\mathbf{p}}_\mathbf{t} &= \mathbf{u}\overline{\mathbf{x}}_\mathbf{1} + \mathbf{v}\overline{\mathbf{x}}_\mathbf{2} + (\mathbf{1} - \mathbf{u} - \mathbf{v})\overline{\mathbf{x}}_\mathbf{3} \\ \mathbf{P}_{\mathbf{p_t}} &= \mathbf{u^2}\mathbf{P}_{\mathbf{x_1}} + \mathbf{v^2}\mathbf{P}_{\mathbf{x_2}} + (\mathbf{1} - \mathbf{u} - \mathbf{v})^\mathbf{2}\mathbf{P}_{\mathbf{x_3}} \end{aligned} \tag{7}$$

3.3.2 From 3D to 1D distribution in space

This section addresses the problem of computing the mean and variance of a point along a 3D line direction if its 3D distribution is known. We consider the example of Figure (2(b)) and focus on computing the segment $\mathbf{p_v} = (\mathbf{p_{v1}}, \mathbf{p_{v2}})$ which indicates with some specified confidence how point $\mathbf{p}$ can move along the ray $\mathbf{r}$. In reality the 3D point $\mathbf{p}$ does not lie exactly on the examined 3D ray $\mathbf{r}$ because it is our estimate of the the true position in the scene. Instead we have an ellipsoid which indicates where the actual point lies in 3D space with some probability. Consider this ellipsoid as a cloud of possible positions for the scene feature and project each of these point onto $\mathbf{r}$. Every point $\mathbf{m}$ on the line can be expressed as $\mathbf{m} = \mathbf{c} + \lambda\mathbf{v}$ where $\mathbf{v}$ is the unit vector along $\mathbf{r}$.

We can estimate the segment $\mathbf{p_v}$ by examination of the distribution of λ for the set of projected points. It can be shown that for each point $\mathbf{x}$ projected, λ will be

$$\lambda = \frac{\mathbf{v} \cdot (\mathbf{c} - \mathbf{x})}{\|\mathbf{v}\|^\mathbf{2}} \tag{8}$$

If we consider equation (8) as a linear transformation of $\mathbf{x}$ to λ then based on equation (1) the mean $\overline{\lambda}$ and the covariance $\mathbf{P}_\lambda$ of λ will be

$$\begin{aligned} \overline{\lambda} &= \tfrac{\mathbf{v}\cdot\mathbf{c} - \mathbf{v}\cdot\overline{\mathbf{x}}}{\|\mathbf{v}\|^\mathbf{2}} \\ \mathbf{P}_\lambda &= (-\mathbf{v}/\|\mathbf{v}\|^\mathbf{2})^\mathbf{T}\mathbf{P}_\mathbf{x}(-\mathbf{v}/\|\mathbf{v}\|^\mathbf{2}) \end{aligned} \tag{9}$$

where $\overline{\mathbf{x}} = \mathbf{p}$ the mean of distribution of points in the ellipsoid and $\mathbf{P_x} = \mathbf{P_p}$ the covariance. Using equation (9) we consider the point $\mathbf{m} = \mathbf{c} + \overline{\lambda}\mathbf{v}$ as the mean point of the segment $\mathbf{p_v}$ and $\mathbf{p_{v1}}, \mathbf{p_{v2}}$ are defined left and right of $\mathbf{m}$ in distance $\mathbf{P}_\lambda$.

For the 3D point $\mathbf{p_t}$ on the triangle we use equation (6) to compute the values of u, v. Its 3D distribution can then be computed using equation (7). Using the same reasoning with $\mathbf{p}$ we compute the 1D variation segment $\mathbf{t_v} = (\mathbf{t_{v1}}, \mathbf{t_{v2}})$ along the optical ray.

Having computing both $\mathbf{p_v}, \mathbf{t_v}$ we then apply visibility according to definition 3.

4 Results

This section presents results of applying the recursive reconstruction algorithm to sparse 3D structure from both real and synthetic image sequences. The sequences are from simple 3D scenes which contain multiple objects such that the entire scene is not visible from a single viewpoint.

Results for a synthetic image sequence of 25 images are shown in Figure 3. The sequence is of two cubes, one in front of the other and three perpendicular planes behind. In the initial frame the smaller cube is completely occluded but as the camera moves from left to right becomes visible. Our reconstructed model (Figure 3(b)) approximates the real scene surfaces closely. However, several peaks are formed from the triangular sets that correspond to most of the real planes.

Our probabilistic integration weights each estimate according to its uncertainty so that the influence of a single measurement does not significantly affect the model. Figure 3(c) presents the model reconstructed with the proposed update process. To measure the improvement over the reconstruction we fitted a plane to each set of points that belong to the same plane in the real scene and compute the mean and the variance of distances. A comparison of the plane measurements for the new scheme over the equal weight update one is presented in Figure (5a).

Results for a real-image sequence of 10 frames are shown in Figure 4. The scene is of the corner of a room with several occluding objects. The reconstructed models (Figures 4(b), 4(c)) clearly approximate the real scene. Figure (5b) again illustrates the mean and variance of distance of points from their corresponding planes. Although there is an improvement in reconstruction of most planes further evaluation of the possible sources of errors should be made for real scenes.

5 Conclusions

This paper presents a geometric statistical framework for using uncertainty in a 3D indoor environment reconstruction system. In particular we described the formulation for propagating uncertainty through our recursive algorithm for scene reconstruction from sparse data. Results from both real and synthetic sequences demonstrated the importance of explicit uncertainty consideration. The use of uncertainty enables: a)incremental update of the model based on the statistical distribution of features and b)the robust visibility constraint tests to eliminate inconsistent triangles. Further work is required to fully evaluate the reconstruction accuracy and robustness in the presence of measurement uncertainty and feature mismatches in real environment sequences.

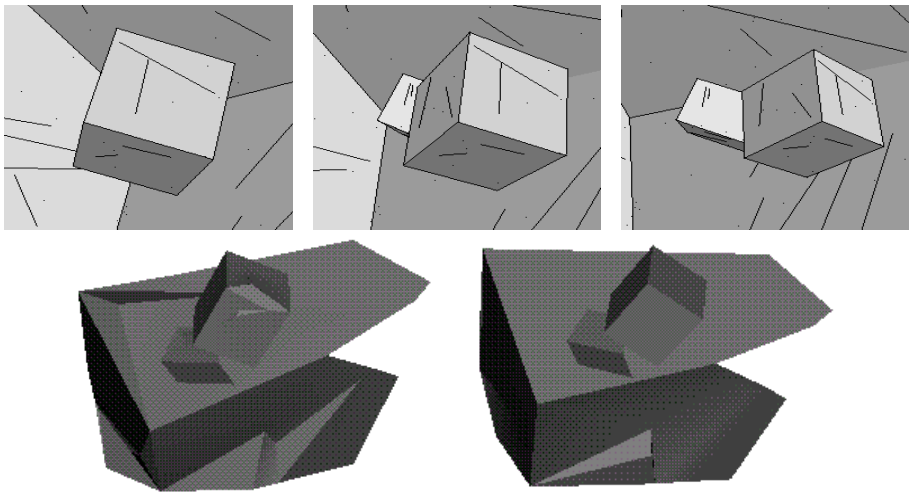

Figure 3: (a)[Top] Synthetic box sequence. (b)[Bottom-Left] Reconstructed model using equal weighs in feature update. (c)[Bottom-right] Model using the statistical weights update.

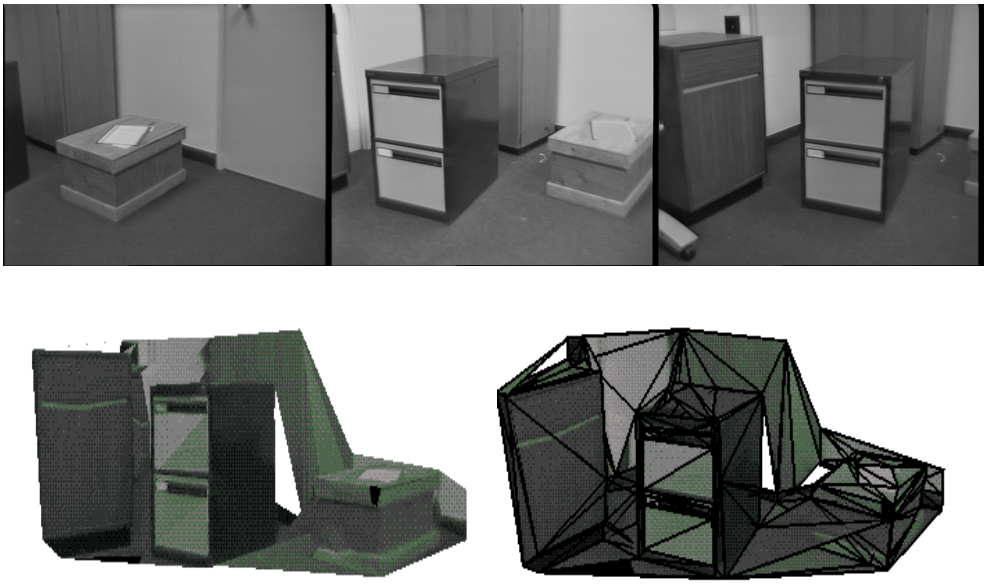

Figure 4: (a)[Top] Real sequence from a room corner. (b)[Bottom-Left] Textured reconstructed model. (c)[Bottom-right] Wireframe superimposed on the model.

Pl.	Stat. weights		Equal weights	
	Mean	Var	Mean	Var
5	0.0002	3.3e-09	0.1414	0.0229
3	9.7e-05	9.6e-10	0.1075	0.0049
0	2.6e-05	2.5e-10	0.0087	3.6e-05
15	0.0051	7.4e-06	0.5277	0.1200
13	0.1695	0.009	0.6770	0.1972
6	0.0001	5.3e-09	0.0325	0.0004
9	0.0001	6.3e-09	0.0021	2.2e-06
17	0.3115	0.2299	0.6218	0.2288

Pl.	Stat. weights		Equal weights	
	Mean	Var	Mean	Var
12	0.0028	4.6e-06	0.0022	4.3e-06
1	0.0117	8.7e-05	0.0146	9.8e-05
0	0.0391	0.0005	0.0569	0.0015
3	0.0852	0.0042	0.0930	0.0037
8	0.0115	0.0001	0.0129	0.0003
11	0.0461	0.0008	0.0508	0.0009
10	0.0336	0.0002	0.0573	0.0013
9	0.0022	2.0e-06	0.0044	9.3e-06
2	0.0610	0.0021	0.2424	0.0601
7	0.0118	0.0001	0.0196	0.0002

Figure 5: Comparison of mean and variance of points from their corresponding planes.(a)Synthetic Sequence. (b)Real lab sequence

References

[1] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[2] S. El-Hakim, C. Brenner, and G. Roth. An approach to create virtual environments using range and texture. In *International Symposium of Real Time Imaging*, pages 331–338, 1998.

[3] O.D. Faugeras, E. Le Bras-Mehlman, and J.D. Boissonnat. Representing stereo data with the delaunay triangulation. *Artificial Intelligence*, 44:41–87, 1990.

[4] S.B. Kang and R. Szeliski. 3-d scene data recovery using omnidirectional multibaline stereo. In *CVPR*, pages 364–370, 1996.

[5] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *ICCV*, 1999.

[6] A. Manessis, A. Hilton, P. Palmer, P. McLauchlan, and X. Shen. Reconstruction of scene models from sparse 3d structure. In *CVPR*, June 2000.

[7] P Maybeck. *Stochastic models,estimation and control*. Academic Press, 1979.

[8] P. McLauchlan. The variable state dimension filter. Technical Report VSSP 4/99, University of Surrey, Department of Electronic Engineering, 1999.

[9] P. McLauchlan and D. Murray. A unifing framework for structure and motion recovery from image sequences. In *ECCV*, pages 314–320, 1995.

[10] P. McLauchlan, X. Shen, A. Manessis, P. Palmer, and A. Hilton. Surface-based structure-from-motion using feature groupings. In *ACCV*, Jan 2000.

[11] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *IEEE Computer*, 22(6):85–90, 1989.

[12] V. Sequeira, K. Ng, E. Wolfart, J.G.M. Gonalves, and D. Hogg. Automated reconstruction of 3d models from real environments. *Photogrammetry and Remote Sensing*, 1999.

[13] R. Smith and P Cheeseman. On the represantation and estimation of spatial uncertainty. *Intersnational Journal of Robotics Research*, 5(4):56–68, 1987.

[14] C. Therrien. *Discrete Random Signals and Statistical Signal Processing.* Prentice Hall, 1992.

[15] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis.* Springer-Verlag, 1992.